AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions and listings of claims in the application.

1. (currently amended) A method for providing content, comprising the steps of:

receiving a request for particular content, said request is received at a server;

accessing a mark-up language description of said particular content, said mark-up language description includes one or more source files which describe a user interface behavior of said particular content;

compiling said mark-up language description of said particular content to create executable code for a rendering entity different than and within a browser, said executable code provides said particular content, said step of compiling is performed at said server in response to said request; and

transmitting said executable code from said server to said rendering entity.

2. (cancelled)

3. (original) A method according to claim 1, wherein:

said executable code implements a user interface that provides access to said particular content.

4. (currently amended) A method according to claim 1, wherein:

said particular content includes data;

said one or more source files define a connection to an external data source for said data, said external data source is external to said server;

said method further includes accessing said data at said external data ~~a~~ source ~~external to said server~~ in response to said mark-up language description; and

said data is compiled to executable code during said step of compiling.

5. (original) A method according to claim 4, wherein:

said step of compiling includes converting said data to action script and compiling said action script into action script byte code.

6. (original) A method according to claim 1, wherein:
said step of transmitting includes using HTTP to transmit said executable code via a network.

7. (previously presented) A method according to claim 1, further comprising the step of:
executing said executable code at said rendering entity.

8. (currently amended) A method according to claim 1, further comprising the steps of:
accessing media content, said particular content includes said media content;
transforming said media content to an accepted format; ~~and~~
providing a reference to said transformed media content in said executable code; and
adding said transformed media content to said executable code.

9. (original) A method according to claim 1, wherein said step of compiling comprises the steps of:
converting said mark-up language description to action script; and
compiling said action script into action script byte code.

10. (currently amended) A method according to claim 9, further comprising the steps of:
accessing media content, said particular content includes said media content;
transforming said media content to an accepted format; and
adding said transformed media content to said executable code, said request is from a ~~said~~ client associated with said rendering entity, said executable code implements a user interface that provides access to said particular content, said particular content includes data and said data is compiled to executable code during said step of compiling.

11. (currently amended) A method according to claim 1, further comprising the step of:

authenticating said request, said steps of compiling and transmitting are only performed if said step of authenticating is successful, different types of authenticating are provided for different types of content and/or for each item of content.

12. (cancelled)

13. (currently amended) A method according to claim 1, further comprising the steps of:

receiving a request from ~~said~~ a client associated with said rendering entity for second content, said particular content includes a first application, said second content includes a second application called by said first application;

accessing a mark-up language description of said second content;

compiling said mark-up language description of said second content; and

transmitting said compiled mark-up language description of said second content to said client.

14. (previously presented) A method for providing content, comprising the steps of:

receiving a request for particular content, said request is received at a server;

accessing first code associated with said particular content, said first code includes a mark-up language description and a scripting language description;

compiling said mark-up language description and said scripting language description to create combined executable code from both said mark-up language description and said scripting language description that implements a user interface that provides access to said particular content, said step of compiling is performed at said server in response to said request; and

transmitting said executable code from said server to a client.

15. (original) A method according to claim 14, wherein:

said request is from said client.

16. (original) A method according to claim 14, wherein:

said particular content includes data; and

said data is compiled to executable code during said step of compiling.

17. (original) A method according to claim 16, wherein:

said step of compiling includes converting said data to action script and compiling said action script into action script byte code.

18. (original) A method according to claim 14, wherein:

said step of transmitting includes using HTTP to transmit said executable code via a network.

19. (original) A method according to claim 14, further comprising the step of:

executing said executable code at said client.

20. (currently amended) A method according to claim 14, further comprising the steps of:

accessing media content, said particular content includes said media content;

transforming said media content to an accepted format; ~~and~~

providing a reference to said transformed media content in said executable code; and

adding said transformed media content to said executable code.

21. (currently amended) A method for providing content, comprising the steps of:

receiving a request for content that includes data other than code, said request is received at a server;

accessing a mark-up language description associated with said content at said server, said mark-up language description defines a connection to an external data source for said data, said external data source is external to said server;

acquiring said data from said external ~~a~~ data source ~~external to and different than said server~~ in response to said mark-up language description, said data is acquired by said server;

compiling said content at said server to create executable code, said content is based on said mark-up language description and said data, said executable code includes a representation of said data, said step of compiling is performed in response to said request; and

transmitting said executable code from said server to a client.

22. (original) A method according to claim 21, wherein:
said request is from said client.

23. (original) A method according to claim 21, wherein:
said executable code implements a user interface that provides access to said data.

24. (original) A method according to claim 21, wherein:
said step of compiling includes converting said data to action script and compiling said action script into action script byte code.

25. (original) A method according to claim 21, wherein:
said step of transmitting includes using HTTP to transmit said executable code via a network.

26. (original) A method according to claim 21, further comprising the step of:
executing said executable code at said client.

27. (original) A method according to claim 21, further comprising the steps of:
accessing media content;
transforming said media content to an accepted format; and
adding said transformed media content to said executable code.

28. (currently amended) One or more processor readable storage devices having processor readable code embodied on said processor readable storage devices, said processor readable code for programming one or more processors to perform a method comprising the steps of:
receiving a request for particular content, said request is received at a server;
accessing a mark-up language description of said particular content, said mark-up language description references a media file;
compiling said mark-up language description of said particular content to create executable

code for a plug-in to a browser, said executable code provides said particular content, said step of compiling is performed at said server in response to said request; and

transmitting said executable code from said server to said plug-in.

29. (previously presented) One or more processor readable storage devices according to claim 28, wherein:

said request is from said browser.

30. (original) One or more processor readable storage devices according to claim 28, wherein:

said executable code implements a user interface that provides access to said particular content.

31. (original) One or more processor readable storage devices according to claim 28, wherein:

said particular content includes data; and

said data is compiled to executable code during said step of compiling.

32. (original) One or more processor readable storage devices according to claim 28, wherein said method further comprises the steps of:

accessing media content, said particular content includes said media content;

transforming said media content to an accepted format; and

adding said transformed media content to said executable code.

33. (previously presented) One or more processor readable storage devices having processor readable code embodied on said processor readable storage devices, said processor readable code for programming one or more processors to perform a method comprising the steps of:

receiving a request for particular content, said request is received at a server;

accessing first code associated with said particular content;

compiling said first code to create executable code for a plug-in to a web client, said

executable code implements a user interface that provides access to said particular content, said step of compiling is performed at said server in response to said request; and

transmitting said executable code from said server to said plug-in.

34. (previously presented) One or more processor readable storage devices according to claim 33, wherein:

said request is from said web client.

35. (original) One or more processor readable storage devices according to claim 33, wherein:

said particular content includes data; and

said data is compiled to executable code during said step of compiling.

36. (original) One or more processor readable storage devices according to claim 33, wherein said method further comprises the steps of:

accessing media content, said particular content includes said media content;

transforming said media content to an accepted format; and

adding said transformed media content to said executable code.

37. (previously presented) One or more processor readable storage devices having processor readable code embodied on said processor readable storage devices, said processor readable code for programming one or more processors to perform a method comprising:

receiving a request for content that includes data other than code, said request is received at a server;

acquiring said data from a data source external to said server, said acquiring is performed by said server;

compiling said data at said server to create executable code for a rendering entity that is separate from a browser but operates within said browser, said executable code includes a representation of said data, said step of compiling is performed in response to said request; and

transmitting said executable code from said server to said rendering entity at a client.

38. (original) One or more processor readable storage devices according to claim 37, wherein:

said request is from said client.

39. (original) One or more processor readable storage devices according to claim 37, wherein:

said executable code implements a user interface that provides access to said data.

40. (original) One or more processor readable storage devices according to claim 37, wherein said method further comprises the steps of:

accessing media content;

transforming said media content to an accepted format; and

adding said transformed media content to said executable code.

41. (previously presented) An apparatus, comprising:

one or more storage devices; and

one or more processors in communication with said one or more storage devices, said one or more processors receive a request for particular content, said request is received at a server, said request is from a client, said one or more processors access a mark-up language description of said particular content and compile said mark-up language description of said particular content to create executable code for a plug-in to a HTTP client, said executable code provides said particular content, said compiling is performed at said server in response to said request, and said one or more processors transmit said executable code from said server to said plug-in.

42. (original) An apparatus according to claim 41, wherein:

said executable code implements a user interface that provides access to said particular content.

43. (original) An apparatus according to claim 41, wherein:

said particular content includes data; and

said data is compiled to executable code during said step of compiling.

44. (previously presented) An apparatus according to claim 41, wherein:

said particular content includes media content.

45. (previously presented) An apparatus, comprising:

one or more storage devices; and

one or more processors in communication with said one or more storage devices, said one or more processors perform a method comprising the steps of:

receiving a request for particular content, said request is received at a server, said request is from a client, said client includes a browser and a rendering engine that is different than said browser but operates within said browser,

accessing first code associated with said particular content at said server,

compiling said first code to create executable code for said rendering engine, said executable code implements a user interface that provides access to said particular content, said step of compiling is performed at said server in response to said request, and

transmitting said executable code from said server to said client.

46. (previously presented) An apparatus according to claim 45, wherein:

said particular content includes data stored at a source external to said server, said accessing first code includes accessing said data at said source external to said server; and

said data is compiled to executable code during said step of compiling.

47. (original) An apparatus according to claim 45, wherein said method further comprises the steps of:

accessing media content, said particular content includes said media content;

transforming said media content to an accepted format; and

adding said transformed media content to said executable code.

48. (previously presented) An apparatus, comprising:

one or more storage devices; and

one or more processors in communication with said one or more storage devices, said one or more processors receive a request for content that includes data other than code, said request is received at a server, said request is from a client, said one or more processors access a mark-up language description and a scripting language description associated with said content at said server and acquire said data from a source external to said server, said data is acquired by said server, said one or more processors compile said mark-up language description and said scripting language description at said server to create executable code, said executable code includes a representation of said data, said compiling is performed in response to said request, and said one or more processors transmit said executable code from said server to said client.

49. (original) An apparatus according to claim 48, wherein:

said executable code implements a user interface that provides access to said data.

50. (previously presented) An apparatus according to claim 48, wherein:

said data includes media content.

51. (previously presented) A method according to claim 21, wherein:

said data is media data.

52. (previously presented) A method according to claim 1, wherein:

said request includes an indication that identifies a type of rendering entity from a group of rendering entities; and

said compiling includes creating said executable code specific for said type of rendering entity in response to said indication.

53. (new) A method according to claim 1, wherein:

said executable code comprises one or more binary files.

54. (new) A method according to claim 1, wherein:

said executable code comprises object code.

55. (new) A method according to claim 1, wherein:

said executable code comprises byte code.

56. (new) A method according to claim 1, wherein:

said one or more source files comprise a view template of a user interface element which is instantiated when said executable code is executed by said rendering entity.

57. (new) A method according to claim 56, wherein:

said one or more source files comprise a view class which supplies default properties, behavior, and child views which the view template instantiates.

58. (new) A method according to claim 1, wherein:

said one or more source files comprise an element which references a media file.

59. (new) A method according to claim 1, wherein:

said one or more source files comprise an element which references a media file that contains a static image.

60. (new) A method according to claim 1, wherein:

said one or more source files comprise an element which references a media file that contains an animation.

61. (new) A method according to claim 1, wherein:

said one or more source files comprise an element which references a media file that contains a movie.

62. (new) A method according to claim 1, wherein:

said one or more source files comprise an element which references a .SWF file.

63. (new) A method according to claim 1, wherein:

said one or more source files comprise an element which references a media file that contains audio.

64. (new) A method according to claim 1, wherein:

said one or more source files comprise an inline definition of formatted text.

65. (new) A method according to claim 1, wherein:

said one or more source files comprise an inline definition of vector graphics

66. (new) A method according to claim 1, wherein:

said one or more source files define a visual appearance of said particular content.

67. (new) A method according to claim 1, wherein:

said one or more source files comprise an element that references a media file external to said one or more source files.

68. (new) A method according to claim 1, wherein:

said one or more source files define a connection to a web service.

69. (new) A method according to claim 1, wherein:

said compiling comprises parsing said markup language description to obtain first and second types of elements, providing said first and second types of elements to first and second compiling modules, respectively, to obtain first and second object code, respectively, and assembling said first and second object code into a single executable.

70. (new) A method according to claim 69, wherein:

said first type of element defines at least one of a visual appearance of said content and a

behavior of said particular content, and said second type of element defines a connection to an external data source for said particular content, said external data source is external to said server.

71. (new) One or more processor readable storage devices according to claim 28, wherein:

said media file contains at least one of a static image, an animation, a movie and audio.

72. (new) One or more processor readable storage devices according to claim 28, wherein:

said media file comprise a .SWF file.